COMP++

An object-oriented successor to MACOS...

January 20, 1998

Dave Redding, Meemong Lee
Jet Propulsion Laboratory, California Institute of Technology

Objectives for future MACOS development

- **Upgrades (beyond mere fixes)...**
  - **Expand library of elements**
  - **Expand physics scope**
  - **Efficient memory management**
  - **Simplify S-MACOS and MACOS/Matlab interfaces**
- **Improved...**
  - **Ease of use, whether as a stand-alone app or embedded Matlab toolbox**
  - **Ease of programming**
    - **» Internal, by developers**
    - **» External, by users**
  - **External access to data**
    - **» All MACOS data in Matlab workspace**
    - **» Other databases**
- **Open architecture**
  - **Distributed development**

Object-oriented architecture

- **We have been experimenting with a new object-oriented MACOS**
  - **Complete rewrite in C++**
- **"Object-oriented" means**
  - **Software objects have "same" properties as physical objects, for a more intuitive interface**
  - **New objects can inherit properties and functions from other similar objects, for more efficient programming**
  - **Data hiding and other techniques make it easier to write modular, expandable code**
  - **Easier for *others* to write MACOS/COMP++ code**
- **"Complete rewrite" means**
  - **Recoding everything in a new structure and a new language**
  - **Revalidating everything**

Is it worth it???

Properties of optical systems

- **Elements**
  - **Type: reflective; refractive; diffractive; reference; return**
  - **Segmentation: Segmented mirror; lenslet array;**
  - **Sequence (must be searched for cornercubes, prisms, …)**
  - **Index of refraction; extinction coefficient; …**
  - **Surfaces**
    - **» Location, direction**
      - **Perturbed by structure, controls**
    - **» Apertures; obscurations**
    - **» Shape**
      - **Conicoid; spherical; flat; general aspheres; anamorphic; …**
      - **Deformed or shaped using polynomials, influence functions, gridded data, XYZ data, …**
        - By design
        - By structures, controls
      -
-

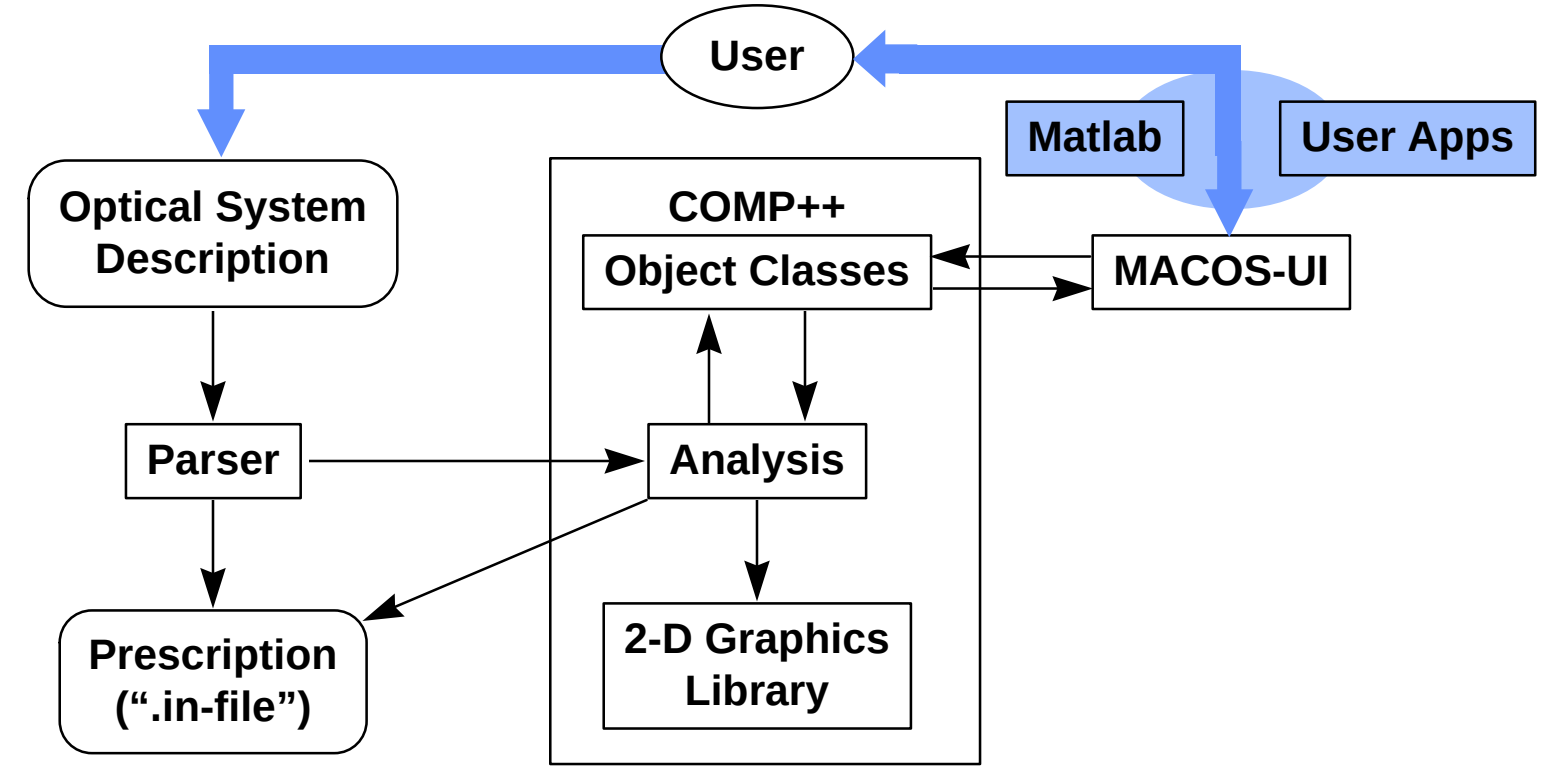
Properties of optical systems (cont.)

- **Sources**
  - **Location, direction, angular extent**
  - **Spectrum (wavelength/flux)**
  - **Phase/amplitude (beam profile)**
  - **Polarization**
- **Beams**
  - **Ray grid**
    - » **Location, direction properties for each ray**
    - » **Number of rays, gridding of rays**
  - **Diffraction grid**
    - » **Mapping to ray grid for geometric properties**
    - » **Amplitude and phase across full beam**
- **Detectors**
  - **Location, direction, angular extent**
  - **Segmentation/pixelation**
  - **QE, crosstalk, noise properties**

Properties of optical systems (cont.)

- **Subsystems**
  - **Super-elements**
    - » **Lenses, prisms, corner cubes, …**
  - **Multiple systems in combination**
    - » **Interferometers, spectrometers, …**
- **More later…**

COMP++ Current Configuration



COMP++ Object Class Configuration

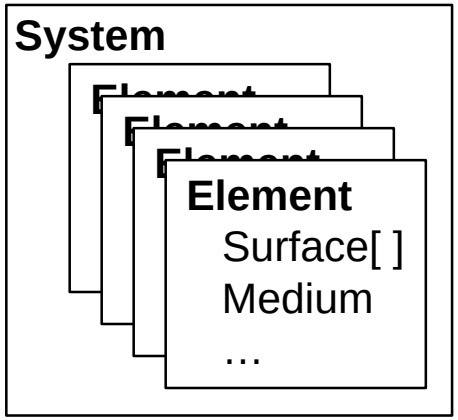
The Light…

Source
Ray
Medium

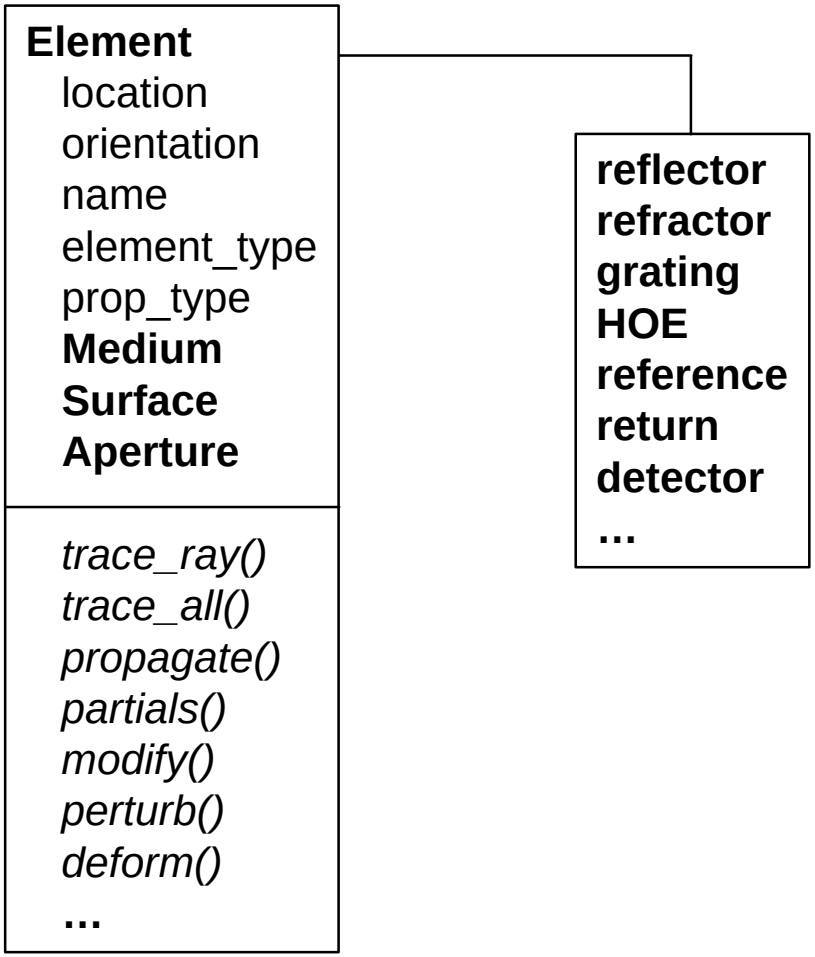
Beam
Ray chief
Ray[ ] rays

Wavefront
Amplitude

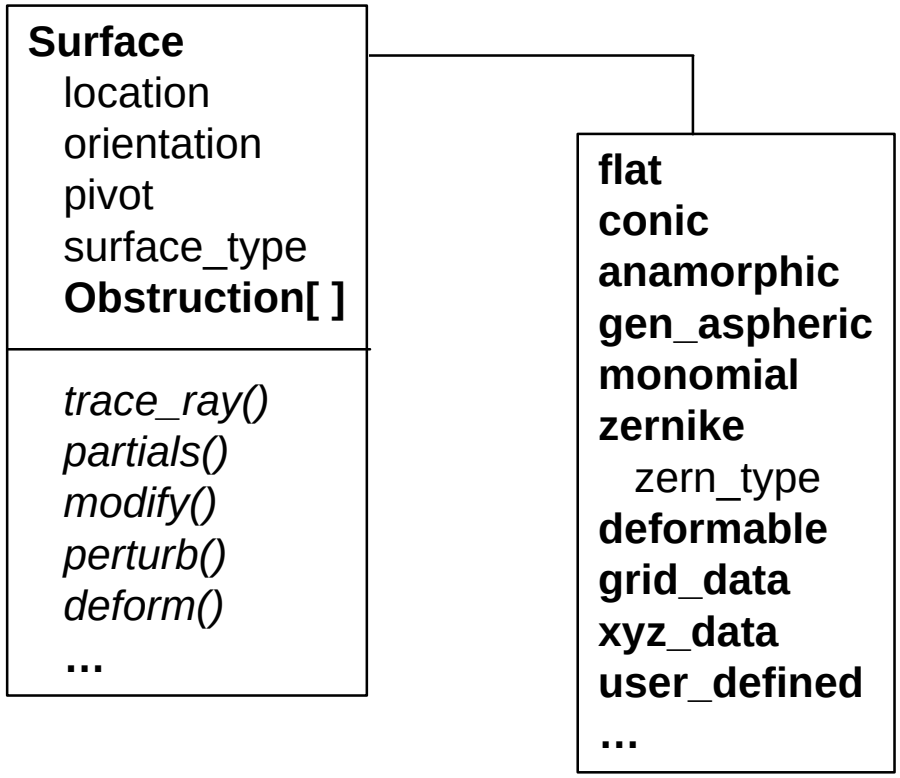
The Optics…



Element Class Definition and Derived Classes



Surface Class Definition and Derived Classes



COMP++ Analysis Configuration

Beam_setup
- optimize_ref_srf
- slave_ref_srf
- center_beam
- set_stop
- find_field_pt
- find_pupil
- perturb
- …

Ray_trace
- trace
- spot_diag
- OPD_map
- find_cross_pt
- …

Propagation
- propagate
- intensity
- read_filter_file
- multispectral_prop
- …

Image_simulation
- compose
- pixilated_image
- noise
- cross_talk
- add_image
- display_composed_image
- stretch
- …

MACOS vs. COMP++: the user perspective

- **Ease of use: stand-alone**
  - **COMP++ better suited to GUI implementation**
  - **Needed?**
- **Ease of use: calling from Matlab or user apps**
  - **COMP++ structures simplify passing of data to and from user application/Matlab workspace, but...**
  - **Same can be done for S-MACOS**
- **Ease of use: programming**
  - **COMP++ provides simpler, safer means for adding new capability**
  - **C++ vs. Fortran**
- **Performance**
  - **COMP++ has efficient memory management**

MACOS vs. COMP++: the developer perspective

- **Ease of use: programming**
  - **COMP++ provides simpler, safer means for adding new capabilities**
- **Cost**
  - **COMP++ requires much work before it equals MACOS capabilities**

Object-oriented IMOS

- **An object-oriented architecture for IMOS as a whole**